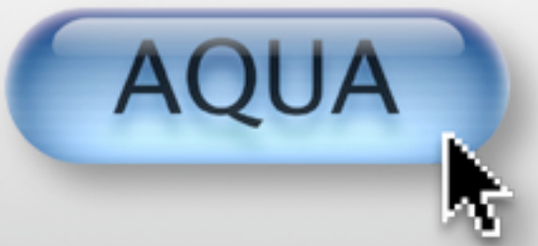
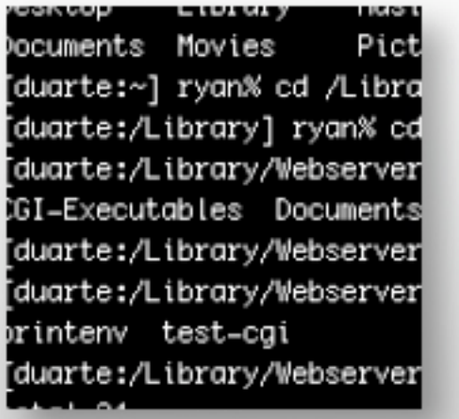
Safari Overview

Internet Technologies

Session 403





Safari Overview

Internet Technologies

Don Melton
Internet Technologies Manager

Introduction

- A high-level overview of Safari technology
- Especially for those developing:
  - Applications
  - Internet tools
  - Web content
  - Apple Help content
  - Open Source

What We'll Cover

- Safari technology you can use in your software
- Other available Internet technologies

Safari

The Safari Story

- Beta 1 released in January
  - Millions of downloads
- Beta 2 released in April
  - Millions more downloads
- 1.0 released this week
  - The servers are busy

Why Is Safari a Success?

- Innovative features
- Solid quality
- Impressive performance

Innovative Features

- SnapBack
- Integrated Google search
- Bookmarks done right
- Pop-up blocking
- Seamless downloads
- Auto-fill forms
- Tabbed browsing
- Bookmarks sync

Solid Quality

- Correctness
  - “It just works”
- Stability
  - “This is just a Beta?”

Impressive Performance

- Fastest browser on Mac OS X
  - Standard benchmarks
  - Loading web pages
  - Launch time
  - Scrolling and resizing windows
- Small memory footprint

How Did We Make Safari So Fast?

- Made performance the priority
- Started with great technology
- Took regular measurements
- Removed sloth before optimizing
- Never made it any slower

Safari Is Swell for Users …

… But What About for Developers?

Not Just a Browser Anymore

Not Just a Browser Anymore

- Moving the technology into public frameworks
- Providing an SDK to developers

What Are These Frameworks?

- Web Kit
  - Provides embeddable “browser” views
- Additions to Foundation
  - Provides upgraded URL APIs

When Are the Frameworks Available?

- Now!
- Built into Panther
- In Jaguar 10.2.x when Safari 1.0 is installed

Using Web Kit

Using Web Kit

Using Web Kit

Using Web Kit

Using Web Kit

Using Web Kit

Using Foundation URL APIs

Using Foundation URL APIs

Using Foundation URL APIs

Using Foundation URL APIs

When Is the SDK Available?

- Now!
- Download using your ADC membership

Where Can I Use the SDK?

- In both Cocoa and Carbon applications
  - Objective-C API
  - Web Kit convenience C API
  - Example C/C++ Web Kit adapter code included

Demo

Richard Williamson
Safari and Web Kit Engineer

http://developer.apple.com/wwdc2003/urls.html

Web Kit Overview

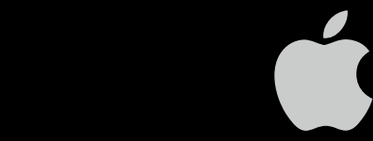
Architecture

Web Technology Layers

Safari

Web Kit

Foundation URL API



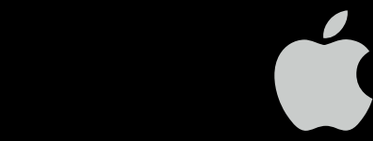
Architecture

Web Technology Layers

Safari

Web Kit

Foundation URL API



Architecture

Web Technology Layers

Safari

Web Kit

WebCore (KHTML)

JavaScriptCore (KJS)

Foundation URL API

At the Core of Web Kit

- Open Source from the KDE project
- WebCore
  - Adapter built on KHTML from KDE
- JavaScriptCore
  - Wrapper around KJS from KDE
- Other Open Source components
  - Expat XML parser, PCRE library, and some Mozilla code

Why Open Source?

- It’s all about the technology
  - Small
  - Fast
  - Easy to modify

Community Collaboration

- Working with KDE team
- Giving back fixes and enhancements
- Sharing ideas and plans for the future
- You can contribute too
  - WebCore and JavaScriptCore
  - KHTML and KJS

Which Frameworks to Use?

- ❌ WebCore and JavaScriptCore
  - Internal frameworks
  - Internal APIs guaranteed to change
- ✅ Web Kit

What Does Web Kit Provide?

- Core classes to display web content in windows
  - HTML, text, images, etc.
- Simplifies complicated loading process
  - Asynchronously requests content from a web server
  - Handles local files too
- Default behavior like a browser
  - Follows links clicked by users
  - Creates and manages all views

Why Use Web Kit?

- Many applications need to display HTML content
- Why write the code to do this yourself?

Standards Support

- Fully buzzword compliant
  - HTML4, XML, CSS, DOM, Cookies, JavaScript, Plug-ins, Java, SSL, Proxies, Unicode, Bidi, etc.
- Real-world compliant
- Working with the W3C on new standards

Basic Web Kit Concepts

- Views
- Frames
- Loading data
- Delegates

Views

- Class: WebView
- Inherits from NSView
- Provides a no-code interface
- With hooks for customization

Frames

- Class: WebFrame
- Every WebView has at least one WebFrame
  - Load requests are sent to main frame
- WebFrame objects created by WebView
  - Can be hierarchy of frames

Loading Data

- Classes: NSURLRequest and NSURLResponse
- Request is URL to load and any options
- Response is everything from server except content

Delegates

- WebFrameLoadDelegate
  - Track changes to what page you're on
- WebResourceLoadDelegate
  - Track loading of resources like images
- WebPolicyDelegate
  - Handle navigation actions like clicks
- WebUIDelegate
  - Manage window and UI-related activity

Demo

Richard Williamson
Safari and Web Kit Engineer

http://developer.apple.com/wwdc2003/urls.html

Foundation URL API Overview

Architecture

Network Technology Layers

Safari

Web Kit

Foundation URL API

CFNetwork

BSD Sockets

Architecture

Network Technology Layers

Safari

Web Kit

Foundation URL API

CFNetwork

BSD Sockets

Architecture

Network Technology Layers

Safari

Web Kit

Foundation URL API

CFNetwork

BSD Sockets

What Does Foundation Now Provide?

- URL loading
- Content caching
- Cookies
- Authentication
- Custom protocols

Why Use New Foundation URL API?

- Features
- Convenience
- Flexibility
- Performance

Convenience

- Smart default behaviors
- Automatic handling of cache and cookies
  - Cache is per application
  - Cookies are shared
- Nearly automatic handling of authentication
  - Just implement two callbacks

Flexibility

- Methods and delegates to control just about everything
  - Request options for reload and offline modes
  - Direct access to cache
  - Multiple cookie policies
- Custom protocols
  - We provide http:, https:, ftp:, file:, and about:
  - You can add your own using the same API

Performance

- Fast URL loading is essential to fast web browsing

Other Internet Technologies

CFNetwork

- All the networking knobs and buttons
- Protocol abstractions
  - Sockets
  - HTTP
  - Rendezvous
- What's new?
  - DNS host resolution
  - FTP

Web Services

- XML-RPC and SOAP support
- What's new?
  - Server-side API

Sherlock

- Easy way to deliver platform-compelling web services
- What's new?
  - Channel browser like Safari bookmarks
  - Third-party channel submission
  - High visibility for your channel
- Improved performance
  - Faster launch and channel switching
  - New XQuery engine 2–3 times faster

Older Technologies

No Longer Under Active Development

- ✗ HTMLRenderingLib
  - ✓ Use Web Kit
- ✗ NSURLHandle
  - ✓ Use new Foundation URL APIs
- ✗ URL Access Manager
  - ✓ Use CFNetwork for some low-level operations
  - ✓ Use new Foundation URL APIs for everything else

Other Older Technologies

No Longer Under Active Development

- ✗ Internet Config
  - ✓ Use Launch Services to ask for helper applications
  - ✓ Use SystemConfig for everything else

Back to Safari

Don’t Forget

- Web Kit can place a “browser” in your applications
- Foundation now has upgraded URL APIs

Wrap Up

Roadmap

102 Open Source at Apple Find out how you can get involved	**Mission** **Tues., 10:30am**
411 Safari Technology and Web Standards Taking advantage of standards in Web Kit	**Nob Hill** **Wed., 10:30am**
418 Advanced Foundation URL APIs High-performance URL loading and more	**Nob Hill** **Thurs., 3:30pm**
420 Advanced WebKit APIs Using HTML rendering capabilities in your app	**Mission** **Fri., 9:00am**

Roadmap

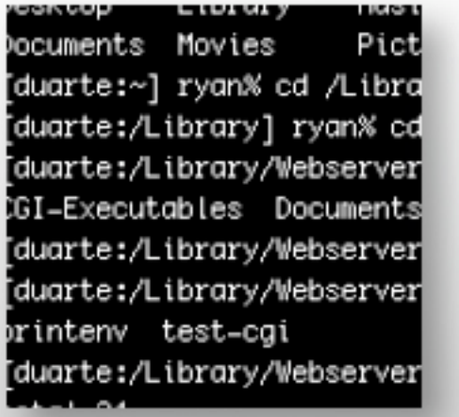
426 Internet Technologies: Delivering Web Services Using Web Channels	**Mission** **Fri., 3:30pm**
110 Overview of Networking APIs High level overview of Networking APIs	**Mission** **Thurs.,10:30am**
112 CFNetwork In Depth All the networking knobs and buttons	**Pacific Heights** **Fri., 5:00pm**
FF015 Internet and Networking Technologies	**Mission** **Fri., 10:30am**

Who to Contact

John Geleynse
Manager, Software Evangelism
geleynse@apple.com

http://developer.apple.com/wwdc2003/urls.html

Q&A



http://developer.apple.com/wwdc2003/urls.html

WWDC 2003

Worldwide Developers Conference